RECEIVED
CENTRAL FAX CENTER
NOV 0 8 2006

Amendments to the Claims: This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A computer system for optimizing processing of an annotation request from a client, comprising:

a request processor configured to receive said annotation request from said client and to break said annotation request down into a plurality of constituent tasks;

a task queue for storing the plurality of constituent tasks that need to be performed for said annotation request;

a thread-controlling means for maintaining a plurality of threads; ~~and~~

an assigning means for assigning said plurality of threads to said plurality of constituent tasks in said task queue; and

task execution means for concurrently executing the plurality of constituent tasks in the respective plurality of threads on the request processor.

2. (Original) A computer system according to claim 1, wherein said plurality of threads is independent from said plurality of constituent tasks stored in said task queue.

3. (Original) A computer system according to claim 1, wherein said plurality of threads is persistent.

4. (Original) A computer system according to claim 1, wherein said plurality of constituent tasks is arranged in a substantially first-in-first-out basis within said task queue.

5. (Original) A computer system according to claim 1, wherein when a thread is available for assignment, said thread is assigned to a constituent task when said constituent task is ready for execution.

6. (Original) A computer system according to claim 5, wherein said assigned thread is released upon conclusion of said constituent task.

7. (Original) A computer system according to claim 1, wherein said plurality of constituent tasks includes checking a cache to determine whether information pertaining to said annotation request is present in said cache.

8. (Original) A computer system according to claim 1, wherein said plurality of constituent tasks includes retrieving information pertaining to said annotation request from one or more sources.

9. (Original) A computer system according to claim 8, wherein said one or more sources include the Internet.

10. (Original) A computer system according to claim 1, wherein said plurality of constituent tasks includes annotating a retrieved web page with additional hyperlinks.

11. (Original) A computer system according to claim 1, wherein said plurality of constituent tasks includes updating a cache with annotated information.

12. (Original) A computer system according to claim 1, further comprising:

a I/O queue for storing a plurality of I/O tasks identified from said plurality of constituent tasks, wherein said plurality of I/O tasks only perform input and/or output functions.

13. (Original) A computer system according to claim 12, wherein two or more of said plurality of I/O tasks are executed in a parallel manner.

14. (Original) A computer system according to claim 12, wherein said task queue is notified upon completion of each of said plurality of I/O tasks.

15. (Previously Presented) A computer system according to claim 14, wherein upon said notification one or more of said plurality of constituent tasks which require results from said completed I/O tasks are rendered ready for execution.

16. (Currently Amended) A computer system for optimizing processing of an annotation request, comprising:

a request processor configured to break said annotation request down into a plurality of requisite tasks needed to execute said annotation request;

a task queue for storing the plurality of requisite tasks needed to execute said annotation request; and

a thread-controlling means for controlling a thread pool having a plurality of threads;

wherein said thread-controlling means assigns an available thread from said thread pool to an execution-ready requisite task and the request processor executes the requisite task in the assigned thread concurrently with the execution of other tasks in other threads.

17. (Original) A computer system according to claim 16, said thread pool is independent of said plurality of requisite tasks.

18. (Original) A computer system according to claim 16, wherein said assigned thread is released back into said thread pool for subsequent assignment when the execution of said execution-ready requisite task is completed.

19. (Currently Amended) A method for optimizing processing of an annotation request received from a client, comprising the steps of:

breaking said annotation request down into a plurality of constituent tasks needed to complete the execution of said annotation request;

storing said plurality of constituent tasks into a task queue;

maintaining a plurality of threads assignable to said plurality of constituent tasks; ~~and~~

assigning an available thread to a constituent task when said constituent task is ready for execution; and

executing the constituent task in the assigned thread on a processor concurrently with other threads executing tasks on the processor.

20. (Currently Amended) A method according to claim 19, further comprising the steps of:

identifying a plurality of I/O tasks from said plurality of constituent tasks;

storing said plurality of I/O tasks into an I/O queue, separate from the task queue; and

executing two or more of said plurality of I/O tasks in a parallel manner.

21. (Original) A method according to claim 20, further comprising the step of:

rendering one or more constituent tasks which require results from said executed I/O tasks ready for execution.

22. (Original) A method according to claim 19, wherein said plurality of threads is persistent.

23. (Original) A method according to claim 19, wherein said assigning of said available thread to said constituent task is independent of the nature of said constituent task.